

Online Monitoring

D.A. Petyt Online Workshop - Jan '01

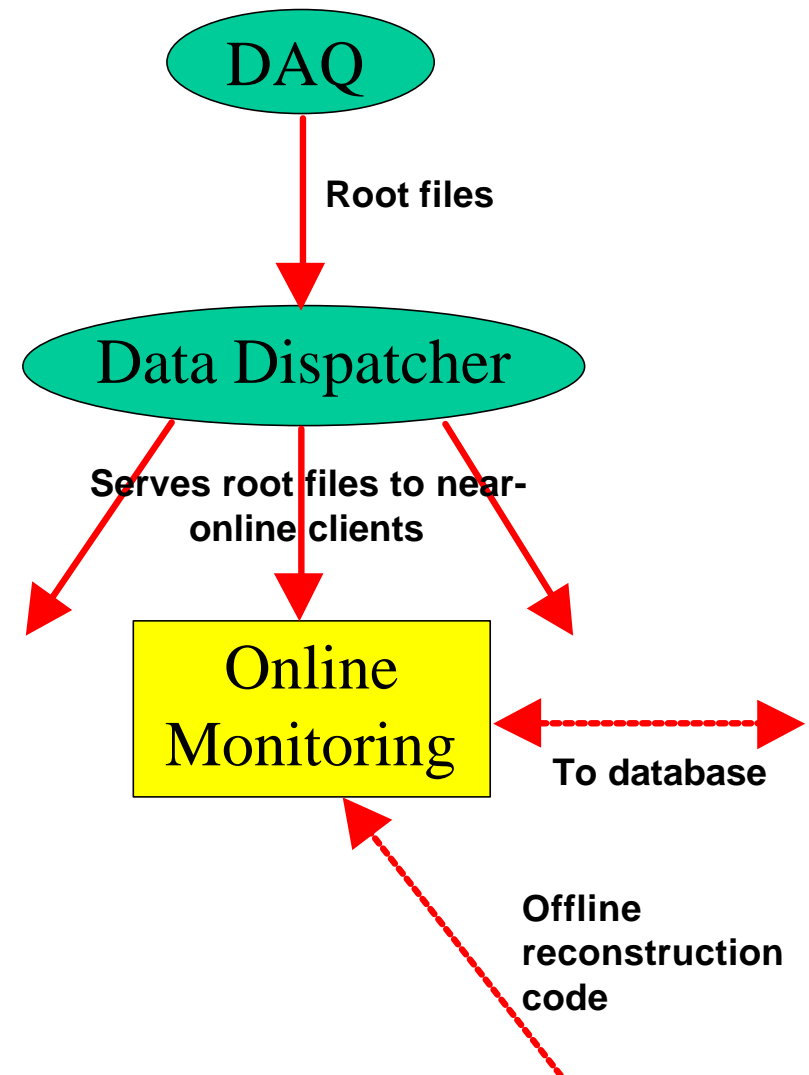
- Purpose of this talk:
 - general overview of monitoring goals
 - description of the monitoring framework
 - list of monitored quantities - variables displayed on-screen
 - errors/actions, logging of output
 - reconstruction needs
- Tuesday/Wednesday session:
 - technical issues, unresolved issues, discussion

Goals of Monitoring

- Online monitoring should:
 - check the performance of the detector elements and track changes on both short and long timescales
 - check the quality of the data
- This is achieved by:
 - constructing channel maps from singles/cosmics to detect hot/dead channels
 - performing simple reconstruction of beam events and cosmics to obtain diagnostic checks of the data (and beam)
- The monitoring system should be able to detect and log failures/anomalies on a timescale of $\ll 1$ hour.
 - A summary containing all relevant data should be produced at the end of each run

Monitoring - Near-Online

- The raw data is written by the DAQ in the form of root files. These contain:
 - beam events, cosemics, singles summaries, flasher summaries
- The online monitoring PC receives this data from the dispatcher on a client/server basis
- The various data types are processed by modules within the monitoring system to produce the monitored quantities
 - this uses reconstruction code developed for the offline system
 - may require constants from the database
- The system runs on its own PC at the detector sites, from which quantities are displayed and logged



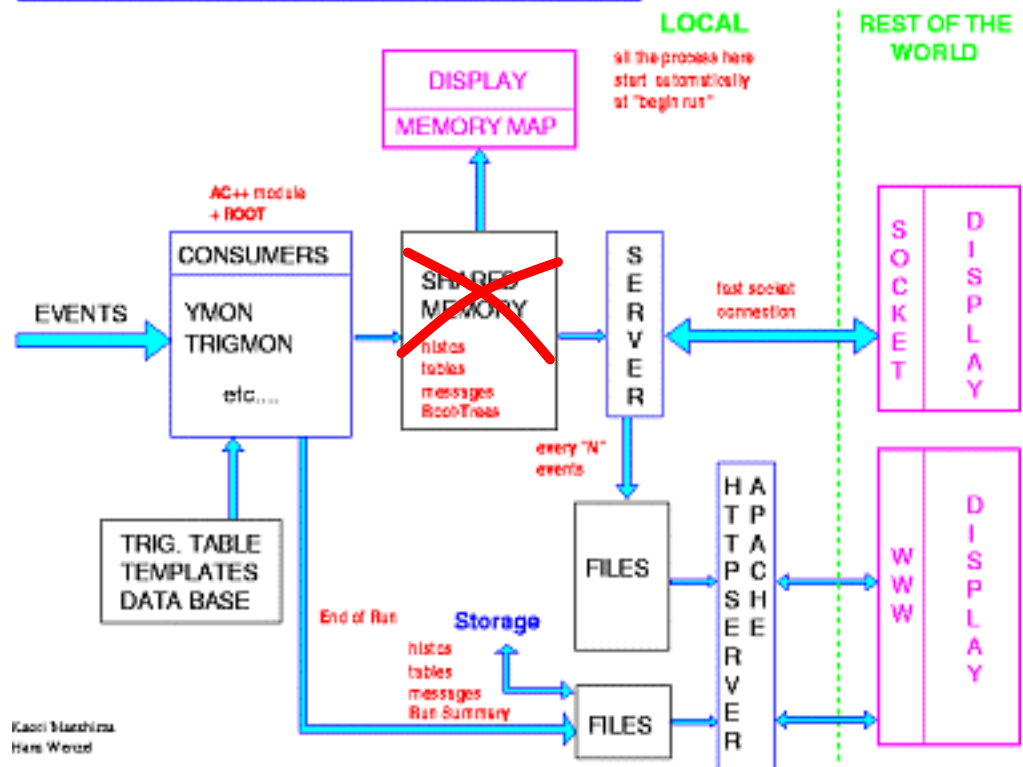
Framework - CDF RunII

- The online monitoring framework is based on the CDF Run II model
- CDF have agreed to supply us with their monitoring code (contact: Kaori Maeshima at FNAL)
- The system is ROOT-based and possesses the features that we require:
 - modular, extensible framework
 - histogram output - display program
 - local/remote access methods
 - + error logging
- We currently have 18 month old test code. New code (developed during 2000 CDF commissioning run is being made available
 - need this a.s.a.p. to begin prototyping

Overview of the framework

- Splitting of Producer, Presenter, Display Server:
- Producer:
 - modules that analyse raw data, create histograms, perform statistical analyses
- Presenter
 - the display program - real-time updates of monitored quantities
- Display Server
 - monitor results transmitted to networked (local/remote) machines

Schematics of Consumers and Display Server



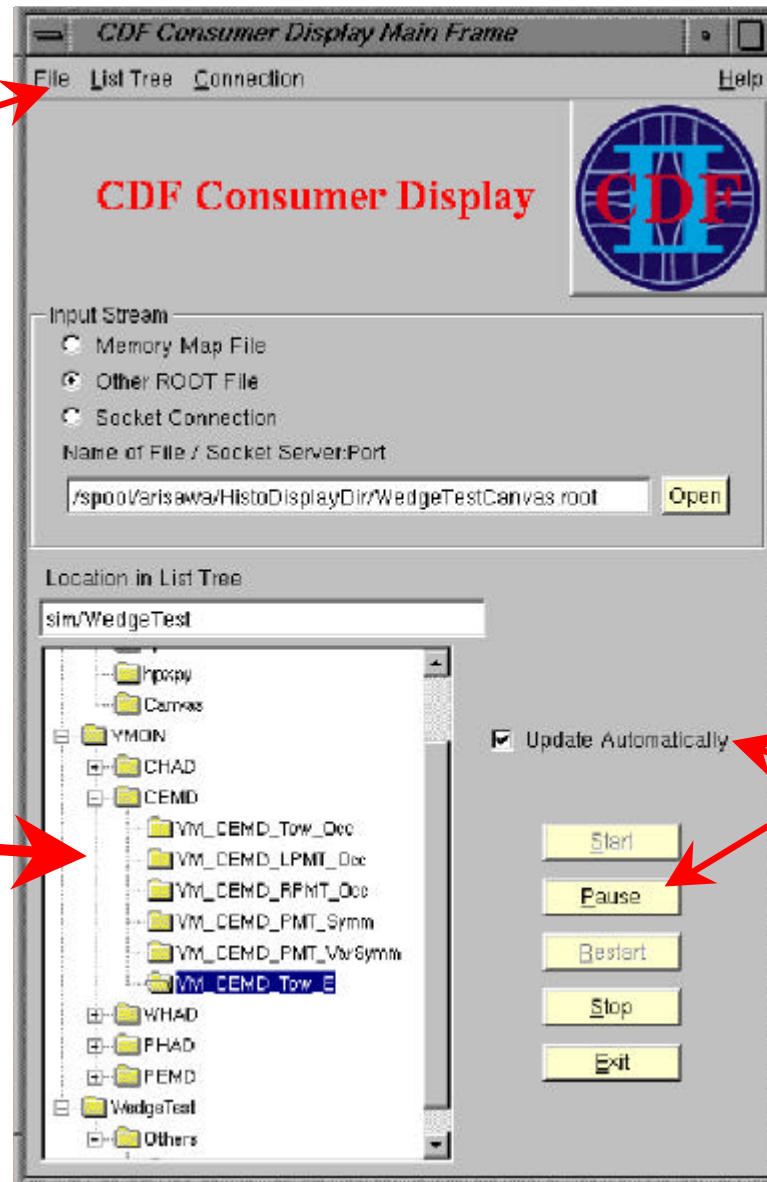
Example of monitoring GUI

Open new GUI/canvases

Select input stream

Click on “Start”

displays List Tree of Modules, sub-processes and variables



Drawing items:

RIGHT CLICK on object in List Tree.

Canvas automatically subdivides if object contains more than one histogram

Deleting items:

LEFT CLICK on object in List Tree and select Delete from List Tree menu

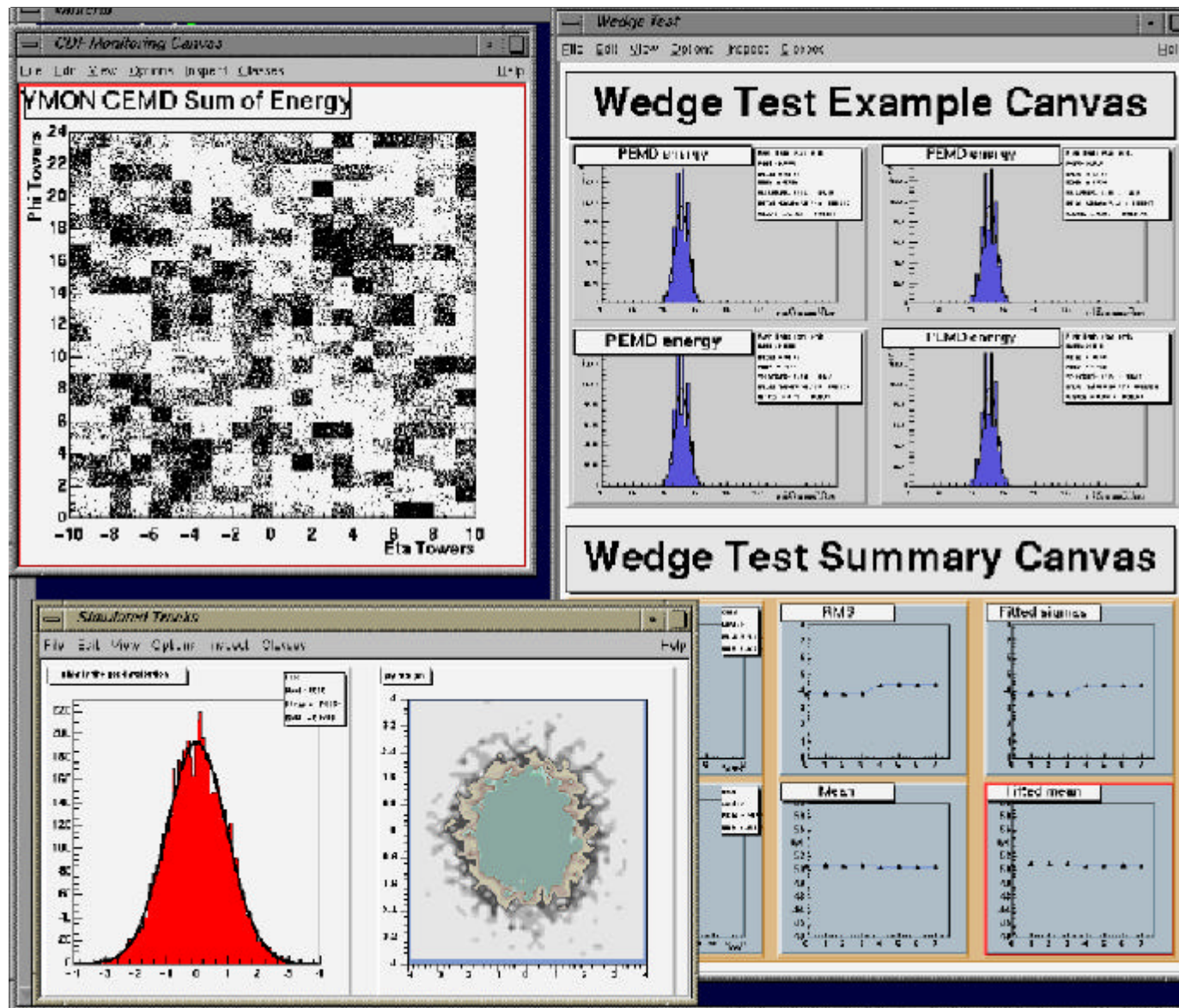
Controls/suspends automatic update of histograms

“ZERO”: resets selected histogram

“RESTORE”: undo of zero command

“CONTENT”: examine bin contents

Example of sample output



Local/Remote Access

- Locally - via root sockets
 - replaces use of memory map file in old framework
 - can allow display program to run off a separate machine from Producer - removes histogram plotting CPU load from monitoring PC (can be a big overhead)
- Remotely -
 - via web access - ROOT-aware Apache server
 - need to write out ROOT file periodically during run
 - via sockets
 - rootd?

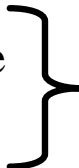
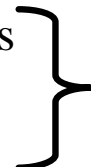

Monitored Quantities

- Several sources:

Data source	NEAR	FAR	CALDET
Beam	<50 Hz	0.1 mHz	80 Hz
Singles	0.6 MHz	1.4 MHz	90 kHz
Cosmics	11 Hz	1 Hz	11 Hz

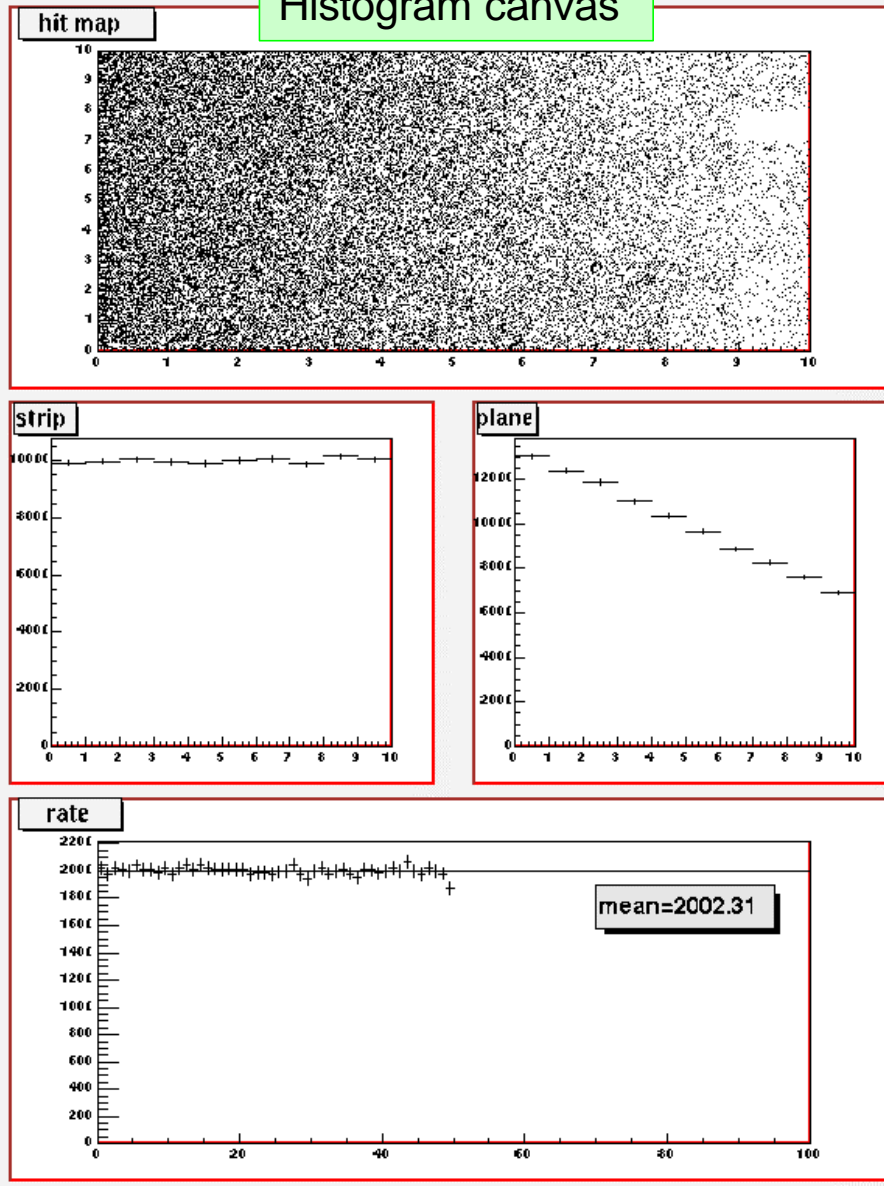
- Usefulness of each source depends on rate and number of channels in each detector:
 - 22 K channels(far), 11K channels(near), 1440 channels(caldet)
 - cannot monitor beam events in far detector
 - cosmics cannot be used to construct real-time hit-maps in far detector

1) Singles rates/hit maps

- The performance of detector elements in each of the three detectors will be principally monitored using singles data
 - rates of cosmic muons are too low, especially in far detector
 - possible longer-term cosmic detector monitoring in near/CalDet
- Trigger farm writes out singles data summaries
 - singles “hits” do not go beyond farm processors
- From these summaries we want to monitor:
 - Rates:
 - overall detector rate
 - per crate rate Means, rms
 - Maps:
 - per channel maps or per pmt maps
 - depends on #channels
 - is per pmt sufficient? Farm rate issue
 - Anomalies
 - list of dead/hot channels Threshold set in Farm

Toy singles rate display

Histogram canvas



Statistics canvas

Number of events = 100000
Time for run = 50.0034 secs
Mean hits/channel = 1000
Rms hits/channel = 194.749
Mean rate/channel = 19.9986 Hz

Simple mock up

- real-time updates of hit map and projections
- real-time rate monitor with running mean
- statistics summary canvas

Issues identified:

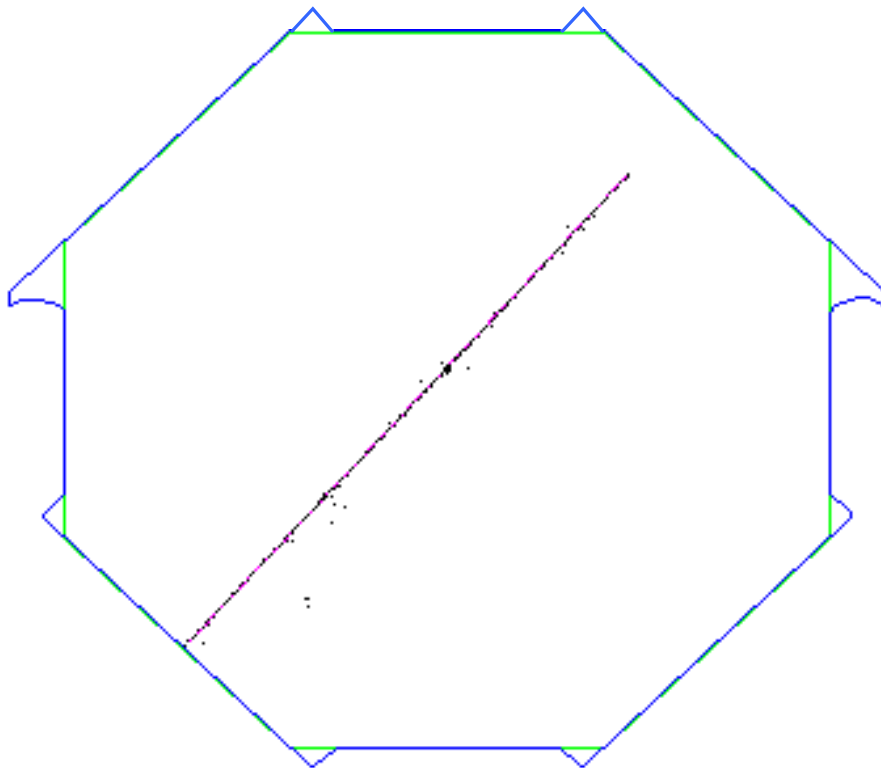
- how often to update (overhead)
- what is the best way of displaying information from large numbers of channels

2) Cosmics/Beam data

- A fraction of the beam data and/or cosmic events will be reconstructed and analysed to monitor the data quality
 - precise fraction to be determined (depends on performance of reconstruction algorithms)
 - current goal is to reconstruct 1000 events/run to obtain sensible statistics on distributions
- Far detector
 - use Paul's cosmics finder or equivalent
 - too few beam events to do anything other than monitor rate
- Calibration detector:
 - reconstruct beam events and cosmics
- Near detector:
 - analyse time frames - most testing reconstruction challenge

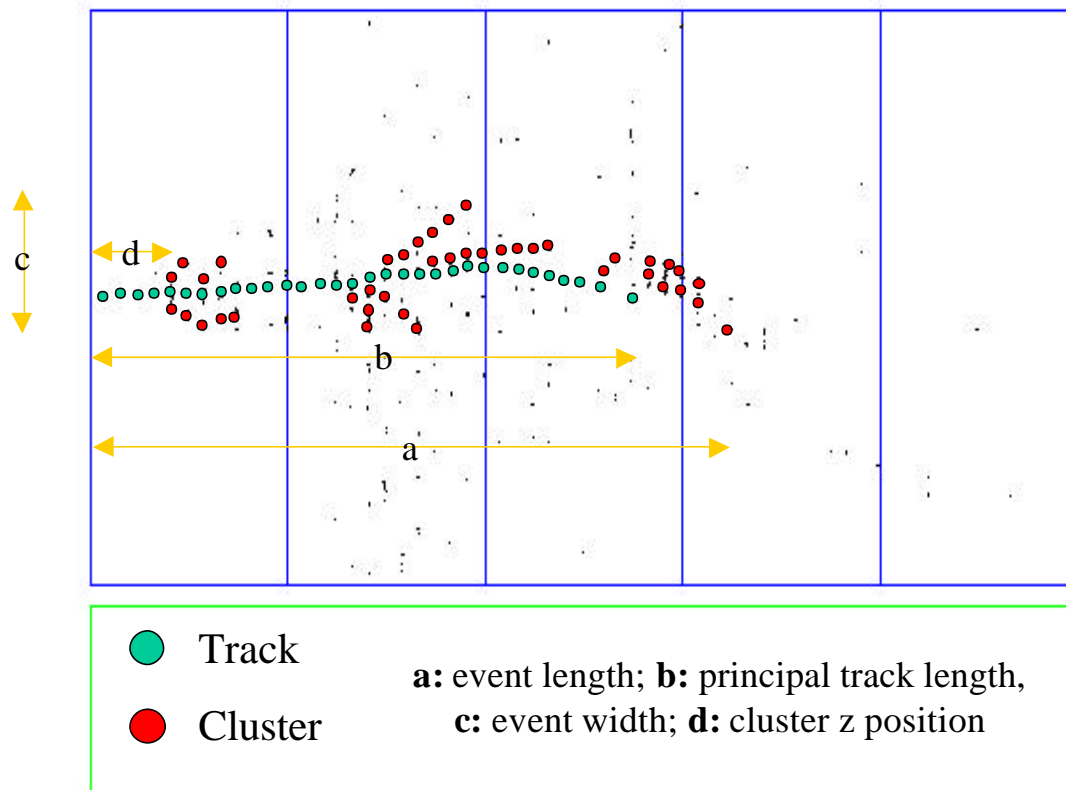
Far detector events

Reconstruction of ~1000 cosmic muons



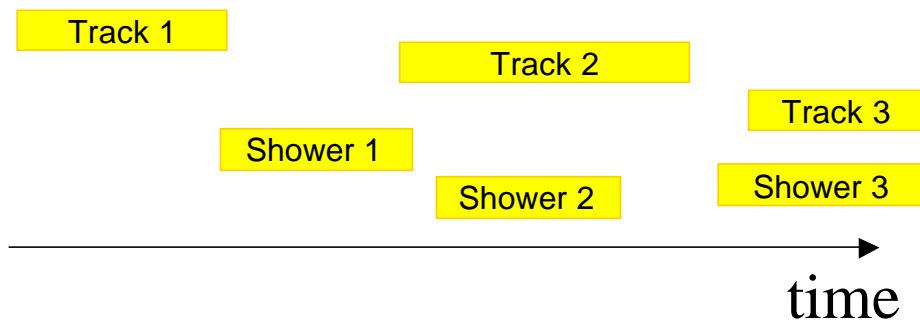
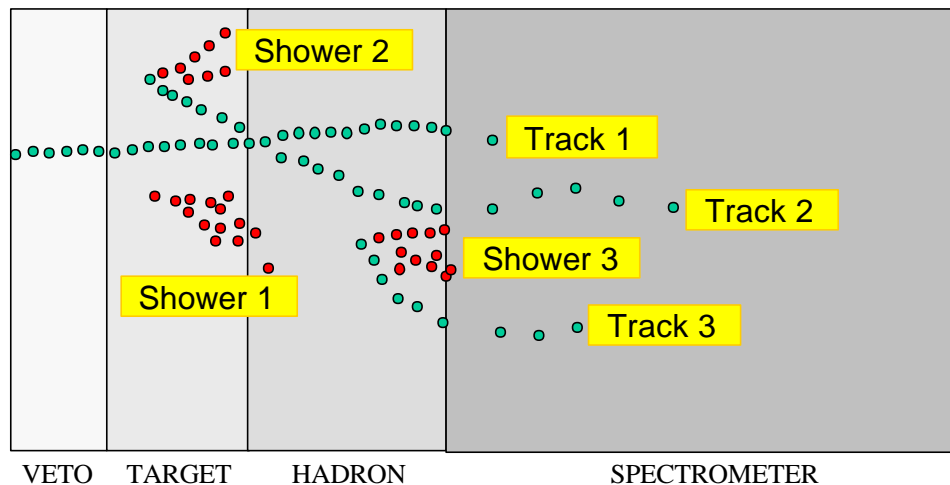
- Number of identified cosmics (efficiency)
- Number of hits
- Mean corrected pulse height
- Track length
- Number of planes crossed
- Track direction cosines
- Track residuals
- Track quality - χ^2 ?
- Plane hit map

Calibration detector events



- Total number of hits
- Summed (corrected) pulse height
- Event flavour: e, μ , h
- Event length
- Event width
- # of track segments
- Principal track: track length, track angle
- # of clusters
- Principal cluster
 - summed pulse height
 - cluster z position
 - cluster angle
- spatial distribution of clusters
- contained/uncontained:
 - 0: contained, 1: exit back,
 - 2: exit side, 3: exit side and back
- Hits per plane

Near detector ~~events~~ time frames



- Total number of hits
 - full detector, region by region, cumulative plane map
- Number of clusters (event candidates)
- Vertex distribution of clusters
- Time distribution of clusters
- # of overlapped clusters
- # of tracks per cluster
- cluster lengths
- cluster RMS widths
- cluster flavour tags
- # of track segments/time frame
- # of hits (+p.h.) in primary cluster
- # of hits (+p.h.) in all other clusters
- contained/uncontained flag

Offline reconstruction code required

- Far detector:
 - cosmic muon finder (Paul Miyagawa code)
 - de-multiplexing code (Brian Rebel code)
- Calibration detector:
 - Track finding Hough, Kalman filter
 - Track fitting Kalman, ...
 - Clustering fast clustering algorithm
 - cosmic muon code Paul's code
- Near detector (as calibration detector) +
 - event splitting Fast splitter (TDC cuts+...)
 - vertex finding
- General offline code:
 - raw data format Robert's classes/code
 - database access Nick's code

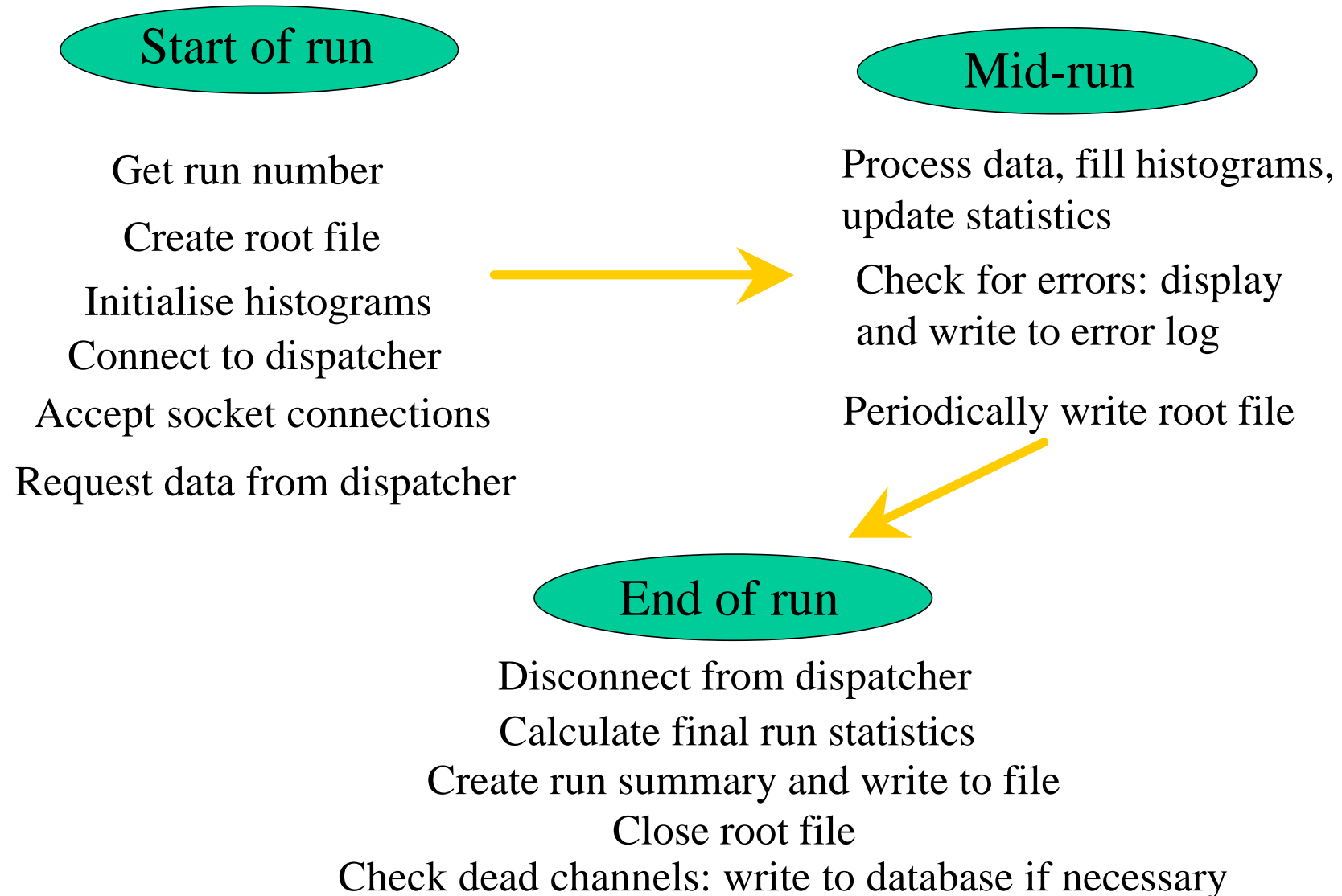
Display/Logging/Interfaces

- Operator display - most pertinent information (flexible):
 - Singles window:
 - total singles rates/ per crate rates
 - total rate as a function of time
 - PMT/channel maps, projections
 - Cosmics/Beam data windows:
 - user-defined views from module List Tree
 - Status window
 - Run number, time, overall rates
 - Message window
 - general messages, error reports
 - GUI window
 - Windows from external sources
 - flasher data

Logging contd.

- Root file (with all histograms) should be saved at end of run. A digest of this information will be generated and saved in a summary file, including:
 - means/rms of important distributions - refer to variable list
 - overall rates
 - append error log (see below)
- Error logging - what can go wrong?
 - dead/hot channels - auto-generated list, should go in database
 - Dead regions of detector - log and raise a flag **!!!!**
 - Anomalously high background rates - raise a flag - should be spotted in DAQ? **!!**
 - No reconstructed beam events **!!!!**
 - Anomalous data distributions
- Error logging model:
 - record (on screen and in error log→database) - no further action?
 - is this enough? - need to ensure all bases are covered

Sequence of events



Next steps

- Obtain new framework and begin tests
- Move over to new Raw Data format
- Integration tests:
 - root file->Dispatcher>Monitoring
- Finalise list of monitored quantities and preferred displays
- Work with reconstruction group to develop/adapt algorithms for monitoring use

Discussion Issues: Tue/Wed

- Monitored quantities is current list complete?
- Rate issues processing power/DAQ rates
- Reconstruction code who/what/when(/how)?
 - requirements
 - interfaces
 - issues
- Logging what, stored where?
- Fault handling who should know/when?
- Local/Remote access methods/security
- Database access what is needed/when?
- Correlation with beam data?